

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

PARKOVACÍ ASISTENT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DAVID MAREČEK

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

PARKOVACÍ ASISTENT

PARKING ASSISTANT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DAVID MAREČEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. DAVID KUBÁT

BRNO 2012

Abstrakt

Tato práce se zabývá návrhem a realizací parkovacího asistenta. Seznamuje s typy senzorů pro měření vzdálenosti a možnostmi využití kamerového systému. Při realizaci je využíváno ultrazvukových senzorů, konkrétně dálkoměrů SRF08 a webových kamer. Také bylo navrženo a implementováno uživatelské rozhraní, které slučuje údaje z jednotlivých senzorů. Parkovací asistent obsahuje funkci pro detekci hran, zvukovou a grafickou signalizaci vzdálenosti spolu s možností automatického nočního režimu.

Abstract

This thesis deals with design and implementation of parking assistant. It introduces the types of sensors for distance measurement and possibilities of using camera system. In implementation there are used ultrasonic sensors, namely rangefinder SRF08 and web cameras. User interface that combines data from individual sensors was designed and implemented. Parking assistant provides function for edge detection, sound and graphics signalization together with automatic night mode.

Klíčová slova

Parkovací asistent, měření vzdálenosti, kamerový systém, ultrazvuková čidla, SRF08, senzor, OpenCV

Keywords

Parking assistant, distance measurement, camera system, ultrasonic sensors, SRF08, sensor, OpenCV

Citace

David Mareček: Parkovací asistent, bakalářská práce, Brno, FIT VUT v Brně, 2012

Parkovací asistent

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Davida Kubáta. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

David Mareček
15. května 2012

Poděkování

Rád bych poděkoval Ing. Davidovi Kubátovi za odborné vedení práce a Fakultě informačních technologií VUT v Brně za poskytnutí senzorů a kamer potřebných pro práci.

© David Mareček, 2012.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Teoretický rozbor	4
2.1	Parkovací asistent	4
2.1.1	Aktivní parkovací asistent	4
2.1.2	Pasivní parkovací asistent	4
2.2	Senzory pro měření vzdálenosti	6
2.2.1	Optické senzory	6
2.2.2	Indukční snímače	8
2.2.3	Kapacitní snímače	8
2.2.4	Ultrazvukové snímače (sonary)	9
2.3	Kamerový systém	11
2.3.1	Barevné modely	11
2.3.2	Obrazové snímače	12
2.3.3	Analogové vs. digitální kamery	13
2.3.4	Kamery	13
2.3.5	Detekce hran v obraze	14
2.3.6	Možnosti přisvícení	15
3	Navrhované řešení	16
3.1	Senzory pro měření vzdálenosti	16
3.1.1	Optické senzory	16
3.1.2	Ultrazvukové senzory	17
3.1.3	Ultrazvukový senzor SRF08	17
3.2	Kamery	20
3.2.1	Centrální kamera	20
3.2.2	Postranní kamery	20
3.2.3	Možnosti přisvícení	21
3.3	Uživatelské rozhraní	21
3.3.1	Prvotní návrh uživatelského rozhraní	22
3.3.2	Finální uživatelské rozhraní	22
3.4	Schéma komunikace	24
3.4.1	FITkit	24
3.4.2	Sběrnice I2C	24
3.4.3	Sběrnice USB	25

4 Implementace	26
4.1 Použité prostředky	26
4.1.1 Qt	26
4.1.2 OpenCV	27
4.1.3 Libfitkit	27
4.1.4 Libkitclient	27
4.2 Zapojení	28
4.2.1 Zapojení SRF08	28
4.2.2 Připojení SRF08 k FITkitu	28
4.2.3 Kamery	28
4.3 Práce se senzory SRF08	29
4.3.1 Práce se sběrnici I2C	29
4.3.2 Inicializace SRF08	29
4.3.3 Start měření	30
4.3.4 Měření vzdálenosti	30
4.3.5 Adresa senzoru	30
4.3.6 Tisk hodnot do terminálu	30
4.4 Přenos dat z FITkitu do QT	31
4.5 Kamery v QT	31
4.6 Popis hlavního programu	31
4.6.1 Práce s kamerovým systémem	31
4.6.2 Detekce hran	32
4.7 Popis uživatelského rozhraní	32
4.8 Přenositelnost	33
5 Testování a další vývoj	34
6 Závěr	36
A Obsah DVD	40

Kapitola 1

Úvod

Ruku v ruce se zvyšujícím se počtem osob na naší planetě stoupá i počet jimi používaných dopravních prostředků. Již v dnešní době je ve většině měst značný nedostatek parkovacích míst, především pro osobní automobily, proto se je jejich majitelé snaží vměstnat do míst, kde by to ještě před pár roky vzdali jako zhora nemožné. Vlivem tohoto snažení, špatného odhadu a nepozornosti dochází k častým nehodám a v hustěji obydlených městech již snad není jediný řidič, který by při podélném parkování ještě, jak se lidově říká, nepostrčil některý ze sousedních automobilů. Na téma problematiky podélného parkování byl dokonce mezi řidiči v květnu 2011 proveden průzkum [1], který si vyžádala společnost Ford. Průzkum probíhal v šesti evropských státech, v každém na vzorku minimálně 1000 respondentů. Z něj vyplynulo, že pro většinu řidičů je podélné parkování značně stresující a potřebují na něj v 32 procentech více než jeden pokus. Pro zajímavost v Itálii řidiči uváděli, že považují podélné parkování za stejně stresující jako návštěvu zubaře. Právě z těchto příčin v posledních dobách stoupá obliba tzv. parkovacích asistentů, kteří parkování značně zjednodušují. A právě vytvořením takového parkovacího asistenta se zabývá tato práce.

První kapitola je ryze teoretická. Na úvod se seznámíme s tím, co to vlastně parkovací asistent je, jaké máme typy a možnosti. Podíváme se na typy senzorů pro měření vzdálenosti, principy a vlastnosti těch nejpoužívanějších. Detailněji se zaměříme na optické a ultrazvukové. Seznámíme se s možnostmi kamerového systému a s tím související základní teorií. Uvedeme typy kamer, které připadají v úvahu při tvorbě parkovacího asistenta. S kamerami souvisí i možnost detekce hran v obraze, zde si povíme jaké máme metody a podíváme se blíže na Cannyho hranový detektor.

V druhé kapitole se budeme zabývat návrhem parkovacího asistenta. Zhodnotíme jednotlivé senzory a uvedeme klady a zápory. V této práci bude využíván ultrazvukový senzor SRF08, proto si ho detailněji popíšeme. Dále vybereme jednotlivé kamery a uvedeme jejich parametry. Poté navrhne uživatelské rozhraní – ukážeme si prvotní a finální návrh a povíme si proč vypadá tak jak vypadá. Dále následuje návrh schématu komunikace, kde si povíme něco o využívaných sběrnících a o FITkitu.

Předposlední kapitola se zabývá samotnou implementací. Na začátek uvedeme použité prostředky – budeme využívat knihoven Qt, OpenCV, libfitkit, libkitclient. Poté je uvedeno konkrétní zapojení a práce se senzory. Zde jsou popsány i konkrétní funkce pro práci se senzorem SRF08 a způsob pro přenos dat ze senzorů přes FITkit až do našeho programu. Dozvíme se jak zobrazit obrazy z kamer v našem programu a popíšeme vytvořené uživatelské rozhraní – jeho prvky a funkce.

Na úplný závěr zhodnotíme dosažené výsledky. Uvedeme jakým způsobem byl asistent testován a navrhne možnosti jeho dalšího vývoje.

Kapitola 2

Teoretický rozbor

V této kapitole si postupně projdeme celou problematiku spojenou s technologií parkovacích asistentů. V první části se budeme věnovat samotnému pojmu parkovací asistent. V další podkapitole se seznámíme s nejdůležitějšími senzory pro měření vzdálenosti a na závěr této kapitoly si projdeme tematiku kamerového systému a jeho možnosti dodatečných úprav obrazu.

2.1 Parkovací asistent

Pod pojmem parkovací asistenty, někdy též označovány jako couvací sety, či parkovací senzory, si většina populace povětšinou představuje nějaký podpůrný systém při parkování a v podstatě mají pravdu. Jako parkovací asistent se označuje soustava senzorů, popřípadě i kamer a dalších prostředků, které mají co nejvíce zjednodušit parkovací manévry. Tyto asistenty se však dají rozdělit do několika kategorií. V první řadě bychom je mohli rozdělit na aktivní parkovací asistenty a ty ostatní, označme je pracovně jako pasivní parkovací asistenty.

2.1.1 Aktivní parkovací asistent

Pod pojmem aktivní parkovací asistent rozumíme systém, který nejenže informuje o dění kolem automobilu, ale i se na parkování sám aktivně podílí. Takovýto systém po aktivaci sám vyhledává vhodné prostory pro parkování a při samotném manévru dokonce ovládá řízení a dokáže tak navést vůz do poměrně malého prostoru. Řidič v tomto případě je instruován jaké kroky provést a zůstává na něm pouze ovládání pedálů a řazení rychlostí. Pro představu přesnosti, systém od společnosti Ford udává jako dostatečný parkovací prostor rozestup odpovídající 1,2 násobku délky automobilu [1]. Tyto systémy se převážně využívají při tzv. podélném parkování, tedy do řady podélně stojících automobilů. Tyto typy parkovacích asistentů bývají zpravidla zabudovány přímo z výroby jako volitelné příslušenství některých automobilů.

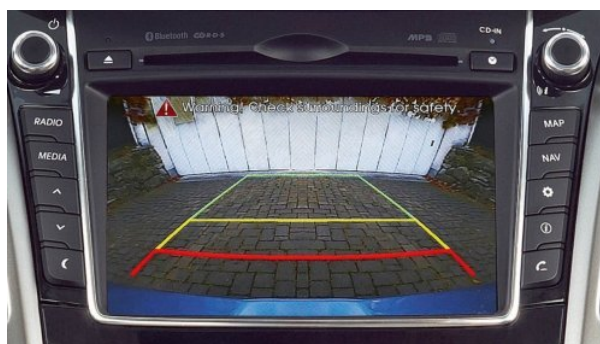
2.1.2 Pasivní parkovací asistent

V první řadě je nutno říci, že tento pojem není nikterak ustálen a v běžné praxi se rozděluje pouze na aktivní a ty ostatní. Avšak principem se jedná o pasivní parkovací asistenty, protože samy do řízení nezasahují. Jedná se tedy pouze o informativní systémy. Tuto samotnou kategorii můžeme ještě rozdělit do menších skupin.

A to například podle způsobu umístění zobrazovacího zařízení (zpravidla LED či LCD displej) ve vozidle, který může být buďto přímo zabudován do palubní desky vozu viz obrázek 2.2. Zde se jedná převážně o parkovací asistenty z automobilů vyšších modelových tříd, kde již jsou přímo součástí palubního počítače. Další možností osazení, asi nejrychlejší a také nejpoužívanější bývají displeje, které se dodatečně umísťují na palubní desku (obrázek 2.3). Třetí možností, která se v praxi vyskytuje je zabudovaný parkovací asistentu přímo do zpětného zrcátka (obrázek 2.1). Parkovací asistenty je také možné najít i jako součást některých GPS navigací.



Obrázek 2.1: Parkovací asistent zabudovaný ve zpětném zrcátku, převzato z [6].



Obrázek 2.2: Parkovací asistent zabudovaný do palubní desky, převzato z [5].



Obrázek 2.3: Nejpoužívanější typ parkovacích asistentů, převzato z [7].

Podle způsobu podávaných informací by se daly asistenty rozdělit na asistenty jednoduché a kombinované. Jednoduché nabízí pouze číselnou hodnotu vzdálenosti k nejbližšímu předmětu, většinou ještě doplněnou o akustickou signalizaci. Kombinované doplňují vzdálenost ještě o obraz z jedné či více kamer.

A poslední parametr, dle kterého bychom mohli dělit, je způsob přenášení dat z čidel, respektive jednotky zpracovávající data ze senzorů, do zobrazovacího zařízení - na drátové a bezdrátové.

2.2 Senzory pro měření vzdálenosti

Měření vzdálenosti je v praxi celkem běžná věc, existuje proto mnoho různých typů senzorů a zařízení, které často bývají velmi úzce specifikovány na použití v určitém prostředí. V této podkapitole se seznámíme s nejpoužívanějšími typy čidel, principy na kterých fungují a jejich výhodami či nevýhodami.[18]

Při výběrů dálkoměrů jsou stěžejní především tyto atributy:

- záběr - tj. jakou oblast je schopen daný senzor pokrýt, udávána ve stupních;
- dosah - do jaké vzdálenosti je možné měřit, udáváno v metrech;
- přesnost - jak je měření přesné;
- chybovost - náchylnost k rušení či odrazům, přeslechům, apod.

Nejčastější přístupy k měření vzdálenosti:

Měření doby letu Založeno na měření doby mezi odesláním a přijetím pulzu, odborně označováno termínem Time Of Flight (TOF).

Měření fázového posuvu Označovány termíny Phase-shift measurement nebo Phase-detection. Založeno na měření fázového posuvu mezi vyslaným a přijatým signálem. Vychází z matematického vztahu: $\varphi = 4\pi \frac{d}{\lambda}$, kde φ je fázový posun, λ je vlnová délka signálu a d je vzdálenost k cíli.

Senzory založené na frekvenčně modulovaném radarovém signálu Založeny na velmi podobném principu jako měření fázového posuvu.

Nejpoužívanějším přístupem je TOF tedy měření doby letu typicky zvukové, rádiové nebo optické vlny.

2.2.1 Optické senzory

Využívají dva základní způsoby měření vzdálenosti k překážce. První způsob je měření doby letu světla od senzoru k překážce a zpět. Druhým pak měření fázového posuvu mezi přijímaným a vysílaným paprskem.

Základní výhody jsou:

1. až tisíckrát rychlejší než ultrazvukové,
2. méně závislý na vlastnostech okolního prostředí.

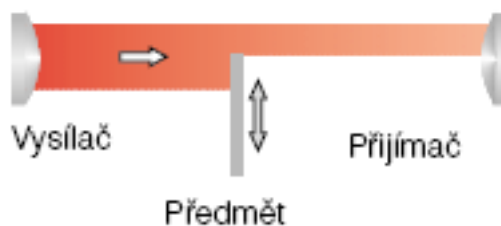
Aplikace přístupů pro měření vzdálenosti na optické senzory

Doba letu TOF Založeno na již zmiňovaném principu měření doby mezi vysláním a přijetím paprsku. Zde využíváme vztah: $t_d = \frac{2R}{c}$, kde R je hledaná vzdálenost, c je rychlost světla, t_d naměřená doba. Nevýhodou této metody je její náročnost na přesnost měření času - vzdálenosti 1 milimetr odpovídá doba letu 6 pikosekund.[30]

Měření fázového posuvu Protože někdy je zajištění přesného a stabilního zdroje času problém, využívá se principu měření fázového posuvu. Pokud je totiž vyslané světlo modulováno, dochází na straně přijímače k posuvu fáze. Tento posuv je pak úměrný času t_d . [30]

Princip optického senzoru/závory

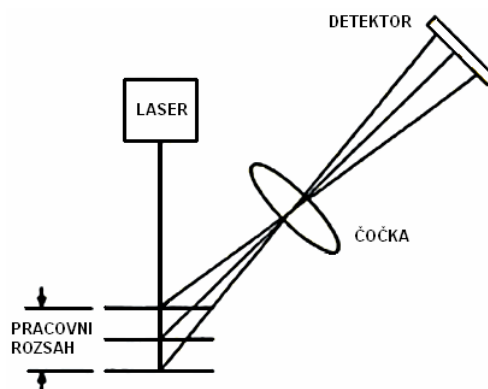
Optické senzory se v praxi velmi často používají v zapojení tzv. závory, kdy jsou proti sobě umístěny vysílač a přijímač (obr. 2.4). Vzhledem k tomu, že toto zapojení je pro nás zcela nevhodné, nebude mu zde věnována větší pozornost. Uvedeno pouze pro ilustraci z důvodu častého využití v průmyslu.



Obrázek 2.4: Funkce optické závory, převzato z [8].

Princip optické triangulace

Laserový paprsek vytváří na detekovaném předmětu světelný bod a ten se odráží. Jak se objekt přibližuje nebo oddaluje, pohybuje se odražený paprsek na detektoru. Ze známých vlastností čočky, detektoru a pozice paprsku na detektoru můžeme vypočítat vzdálenost objektu. Názorný příklad je uveden na obrázku 2.5. Z důvodů různé odrazivosti materiálů se rozlišují dva typy snímačů. Pro lesklé materiály jsou to snímače odrazivého typu a naopak pro materiály, které světlo rozptylují snímače rozptylového typu. [21]

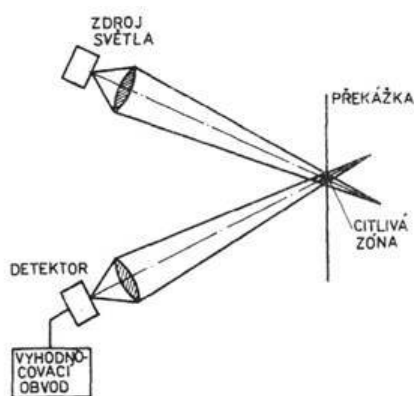


Obrázek 2.5: Princip triangulace, převzato z [21].

Optické senzory založené na intenzitě odrazu

Kužel světla vycházející ze zdroje, usměrněný optikou dopadá na předmět od něho se odráží a tento odraz je optickou soustavou usměrněn na fotocitlivý prvek, který je napojen na vyhodnocovací obvod. Tento princip je názorně zobrazen na obrázku 2.6. Optika vysílače i detektoru je zaostřena do určitého místa, které nazýváme citlivou zónou. Pokud se překážka nevyskytuje přímo v této zóně nedochází k odrazu a tím pádem není co měřit, protože do detektoru žádný paprsek nedorazí. Jako vysílače bývají použity LED diody (dosah jednotek metrů) nebo laserové diody (dosah až kilometry). Jako přijímače jsou využity PIN fotodiody či fototranzistory.

Konkrétně se neměří pouze vzdálenost, ale i úroveň amplitudy nebo světelný výkon. Naměřené hodnoty se poté porovnávají s referenční hodnotou a na základě toho můžeme zjistit i další vlastnosti jako jsou kontrast a barva.



Obrázek 2.6: Princip optického snímače, převzato z [19].

Nejdůležitějším parametrem těchto senzorů je tzv. směrová charakteristika tj. plocha uvnitř které bude předmět detekován. Tento parametr je také závislý na schopnosti odrazu světla od překážky. U překážek s horší schopností odrazu je takto snímaná plocha výrazně menší či dokonce nulová.

2.2.2 Indukční snímače

Slouží pouze pro detekci vodivých kovových předmětů. Měření funguje na principu indukování elektromagnetického pole snímačem, do kterého, když vstoupí objekt z kovového materiálu, tak ve snímači začnou proudit vířivé proudy a narůstá tak zatížení snímače. Naopak při vzdalování zatížení klesá. Na základě tohoto jevu je možné odvodit vzdálenost. Jejich hlavní nevýhodou je malý rozsah, nemožnost detekce nekovových materiálů a jelikož má každý kov různý vliv na elektromagnetické pole tak ani zde není měření vždy zcela jednoznačné [21].

2.2.3 Kapacitní snímače

Fungují na principu generování elektrického pole a snímání jeho kapacity. Avšak je zde základní požadavek na zachování konstantních hodnot měřeného prostředí tj. i velikost a materiál cíle. Při zachování konstatovaných hodnot zůstává jedinou proměnnou vzdálenost

a ta je poté úměrná změnám kapacity. Z důvodů jejich požadavků na konstantní prostředí jsou pro univerzální měření vzdálenosti zcela nevhodné.

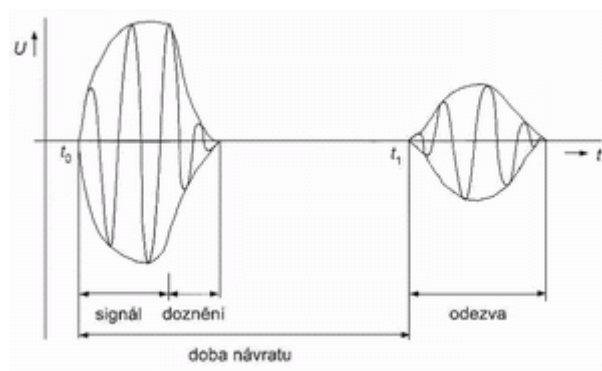
2.2.4 Ultrazvukové snímače (sonary)

Ultrazvuk

Ultrazvuk je definován jako akustické vlnění o frekvenci nad hranicí slyšitelnosti. Za horní hranici ultrazvuku se obecně bere 1GHz, takže se pohybujeme v pásmu 20kHz až 1GHz. V oblasti senzorů pro měření vzdálenosti se pak využívá signálu o kmitočtu 40kHz. Pro ultrazvuk platí stejné fyzikální zákony jako pro jiné zvukové vlny. Jelikož jde o vlnění, které se šíří pouze hmotou není možné jej využívat ve vzduchoprázdnu.[24]

Princip

Měření je založeno na principu měření doby mezi vysláním a přijetím ultrazvukového signálu (TOF). Samotný princip vypadá v obecné rovině následovně: řídicí jednotka vyšle zahajující impuls, elektrický obvod vybudí hliníkovou membránu ke kmitání, tím dojde k vyslání ultrazvukových vln. Vlny dorazí k překážce, část se odrazí a ta poté rozkmitá membránu na přijímacím prvku. Tyto kmity jsou vyhodnoceny. Doba letu a rychlost zvuku v daném prostředí (většinou vzduch) jsou pak dostačující informací pro výpočet vzdálenosti.



Obrázek 2.7: Princip funkce ultrazvukového senzoru, převzato z [24].

Detailnější pohled dle [18] na funkci senzorů je patrný z obrázku 2.7. V čase t_0 je vyslána sada impulsů. Impulsy se šíří prostředím rychlostí zvuku. Dojde-li k nárazu na překážku, část se odrazí zpět, v době t_1 se vrátí k senzoru, zde je detekována. Vyhodnocovací obvody vypočítají vzdálenost z doby zpoždění $t_1 - t_0$.

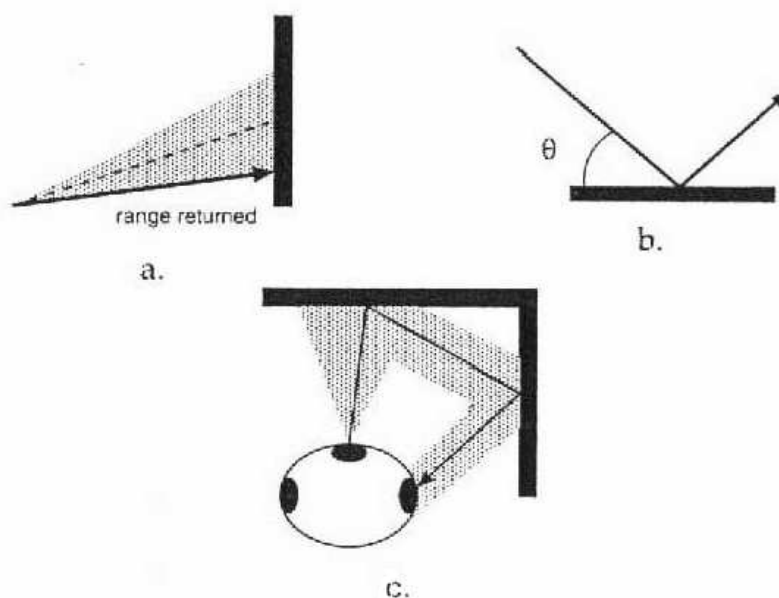
Faktory ovlivňující ultrazvukové senzory

Jelikož je ultrazvuk zvuková vlna, hlavním ovlivňujícím faktorem je prostředí, kterým se šíří, v našem případě tedy vzduch. Rychlost zvuku ve vzduchu je ovlivňována především teplotou, kdy při 0°C má hodnotu 331,6 m/s, ale při teplotě 20°C již 343,8 m/s. Dalšími možnými činiteli mohou být sluneční svit, který senzor zahřívá či zvýšené proudění vzduchu mezi překážkou a senzorem. Tyto činitelé pak mohou mít vliv na případné zkreslení výsledných hodnot.

Nedostatky

Jako největší nedostatek ultrazvukových senzorů se jeví jejich citlivost na tzv. zrcadlový odraz, který vzniká při odrazení zvukové vlny od hladkého povrchu v ostrém úhlu, kdy se vlna odráží místo k senzorům směrem od senzorů viz obrázek 2.8b. Naštěstí v reálném světě většina materiálů obsahuje nějaké hrboly, takže alespoň k minimálním odrazům dojde. Je-li přijímací část sonaru není schopna rozeznat, zda zachycený odraz pochází skutečně z vlastního či cizího zdroje, tak může docházet k tzv. přeslechům (obr. 2.8c), kdy je zachycen cizí odraz. V reálných podmínkách může dojít ještě k tzv. zkrácení, k němuž dochází z důvodu širokého prostorového záběru, kdy jedna část vysílaného kuželu dorazí k překážce dříve. Většina čidel však počítá s tím, že odraz proběhne z centrální části záběru, může tak dojít k virtuálnímu zkrácení vzdálenosti viz obrázek 2.8a.

Další nevýhodou, týká se však pouze systémů s jedním měničem, je tzv. mrtvý čas, kterým označujeme dobu mezi vysláním signálu a dobu, kdy je možné přijmout odezvu, tedy dobu potřebnou pro dokmitání měniče.



Obrázek 2.8: Chyby ultrazvukových senzorů, převzato z [18].

Výhody

Hlavní výhodou ultrazvukových senzorů je jejich univerzálnost. Umožňují detekovat prakticky libovolný materiál bez ohledů na jeho vlastnosti. Některé modely jsou schopny dosáhnout vzdálenosti i desítek metrů.

Základní parametry ultrazvukových dálkoměrů

Nejdůležitější informace nám dává vyzařovací charakteristika. Krom ní jsou udávány ještě následující parametry:

- Aktivní rozsah - maximální dosah, ve kterém je možné provádět měření.
- Mrtvá(slepá) zóna - oblast mezi senzorem a začátkem snímané oblasti, kde není možné provádět měření.
- Úhel odklonu - udává maximální odklon roviny, který je možné ještě měřit.
- Rychlost přeběhu - jakou rychlostí se může objekt pohybovat, aby jej senzor zachytil.
- Reakční doba - doba mezi vysláním impulsu a aktivováním výstupu senzoru.

2.3 Kamerový systém

V této kapitole se seznámíme se základní teorií související s kamerovým systémem. Povíme si o barevných modelech, o typech snímačů a o rozdílech mezi analogovou a digitální kamerou.

Základní parametry udávané u kamer jsou:

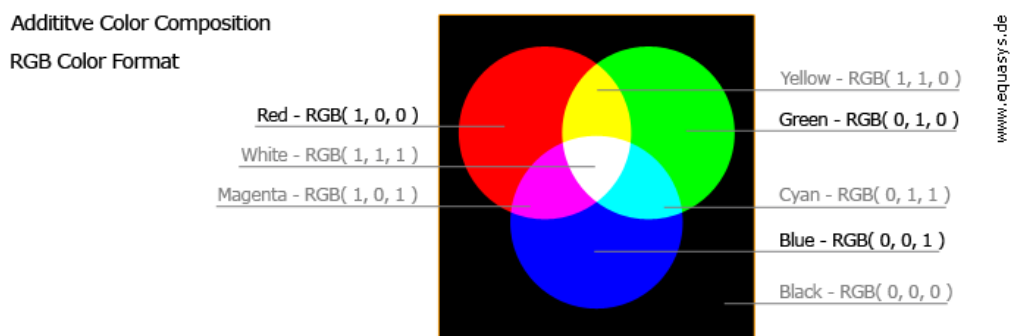
- **FPS** (Frames Per Seconds) označuje počet rámců za sekundu, jinak řečeno kolikrát dojde k zachycení obrazu z kamery během jedné vteřiny.
- **Rozlišení** je udáváno v megapixelech, jedná se vlastně o vynásobení výšky a šířky v obrazových bodech, tedy z kolika milionů bodů je snímek složen.
- **Formát obrazu** - v jakém formátu je dostupný obraz na výstupu z kamery.
- **Možnost připojení** - jakým způsobem je kamera napájena a jak se přenášejí data.
- **Zorné pole** tj. úhel, který je schopna kamera snímat

2.3.1 Barevné modely

Barevné modely se snaží co nejvěrněji napodobit reálné barvy. Určují v kolika kanálech se barvy ukládají a jaké informace jsou v jakém kanálu. Základními modely jsou RGB, CMYK, HSV. Nyní se detailněji podíváme na model RGB, který je nejrozšířenější a na model YUV, který se využívá ve video technice, například pro přenos obrazu formátu PAL.

RGB model

RGB (zkratka red, green, blue) jak zkratka napovídá se skládá ze tří barevných kanálů, tedy červené, zelené, modré. Jedná se o standardní barevný model, založený na faktu, že lidské oko je citlivé právě na tyto tři barvy. Je využíván na všech monitorech. Z toho důvodu se také používá při počítačovém zpracování obrázků či videa. RGB model využívá tzv. aditivního míchání barev. Kdy se jednotlivé složky míchají dohromady a podle intenzity jednotlivých kanálů vzniká požadovaná barva. Jak dostaneme základní barvy je ilustrováno na obrázku 2.9.



Obrázek 2.9: Barevný model RGB, převzato z [16].

YUV model

Tento model využívá tři kanály. Kanál U a V reprezentují barvu a Y určuje její jas. Tento model byl vyvinut z důvodu kompatibility barevného a černobílého vysílání, kdy bylo nutno přidat k jasové složce ještě složku barevnou. Využívá se v televizním vysílání v normě PAL i HDTV. Převod modelu RGB do YUV je možný pomocí vztahu: 2.1 a z YUV do RGB pomocí 2.2, oba vztahy čerpány z [17].

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & 0.515 & -0.100 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (2.1)$$

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1.137 \\ 1 & -0.397 & -0.580 \\ 1 & 2.034 & 0 \end{pmatrix} * \begin{pmatrix} Y \\ U \\ V \end{pmatrix} \quad (2.2)$$

2.3.2 Obrazové snímáče

Obrazový snímáč je odpovědný za převod světla do elektrických signálů. Dopadem světelných paprsků na snímáč (CCD, CMOS) získávají jednotlivé elementy snímáče náboj, který je prezentován napětím a s tím se dále pracuje [25]. Zjednodušeně řečeno slouží k převodu obrazu na data.

CCD snímáče (Charged Coupled Device)

Tyto snímáče jsou nejčastěji používaným typem v dnešních kamerách. Dosahují daleko lepší světelné citlivosti, která se projeví kvalitnějším obrazem. Vzhledem k specifickému výrobnímu procesu jsou dražší a jejich zabudování do kamer je pak složitější.

CMOS snímáče (Complementary Metal Oxide Semiconductor)

Nedosahují takových kvalit jako CCD snímáče, používají se především v místech, kde je rozhodující cena a velikost. Vzhledem k jejich horší citlivosti na světlo se nehodí do tmavších míst.

2.3.3 Analogové vs. digitální kamery

U obou typů kamer dodává obrazový snímač analogový signál. Rozdíl je v jeho následujícím zpracování a zaznamenávání [20].

Analogové kamery

U analogových kamer se signál pouze přizpůsobuje přenosu a dále je přenášen a zaznamenáván v analogové podobě. K samotné úpravě pro zobrazení dochází až na straně zobrazovacího zařízení, v našem případě počítače. Jelikož je obraz přenášen analogově, je zde potřeba speciálního převodníku obrazu. Pokud chceme záznam nějakým způsobem digitalizovat, musíme k tomu být patřičně vybaveni. Proto se tento typ kamer využívá převážně pro úzce specifickou oblast, kde není potřeba další manipulace s obrazem. V dnešní době nacházejí hlavní využití v kamerových systémech, apod.

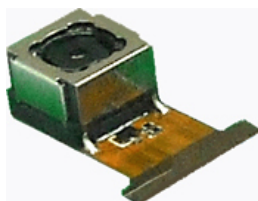
Digitální kamery

U digitálních kamer se obraz z analogové podoby převádí pomocí A/D převodníku na digitální a to rovnou v kameře, to zajišťuje lepší kvalitu obrazu než u kamer analogových. Výstupní záznam můžeme tedy rovnou dále přenášet bez nutnosti specializovaného hardwaru. Oproti analogovým kamerám zde ani při kopírování nedochází ke ztrátám kvality. Tento typ kamer je dnes nejvyužívanější.

2.3.4 Kamery

Kamerové moduly

Jedná se o miniaturní kamery viz obrázek 2.10, většinou bez vnějšího pouzdra, určené k zabudování do jiných zařízení. Jejich hlavní výhodou je jejich miniaturní rozměr, nevýhodou pak může být u modulů s kvalitnějším obrazem jejich vyšší cena. Výstupní data je možné odebírat jak analogově tak i digitálně. Další značnou výhodou je jejich variabilita, kdy můžeme volit například formát výstupního obrazu. Tyto moduly jsou také plně ovladatelné přes sběrnici I2C.



Obrázek 2.10: Příklad kamerového modulu, převzato z [12].

Webkamery

Mezi jejich hlavní výhody patří pořizovací cena a jednoduchá aplikace. Stačí připojit popřípadě nainstalovat ovladače a jsou připraveny k použití. Ihned po připojení, bez dalšího nastavování je možno využít i funkce jako automatické ostření, či záznam zvuku, pokud je tedy těmito funkcemi kamera vybavena. Některé modely jsou pak přímo osazeny i LED diodami pro případné přisvícení. Nevýhodami pak může jejich menší konfigurovatelnost, kdy

některé prvky jsou nastaveny z výroby a není možno je měnit, jako třeba formát výstupního obrazu.

2.3.5 Detekce hran v obraze

Pokud se v obraze snažíme detekovat hrany, hledáme taková místa, kde dochází ke skokovým změnám intenzity. Rozlišujeme dva typy hran a to hrany jasové – v místech, kde dochází k lokální změně jasu a na jasové hraniční elementy – místa, kde dochází ke globální změně. K detekování hran existuje hned několik algoritmů, jedna skupina z nich je založena na první derivaci a druhá na druhé derivaci.

Metody založené na první derivaci

Jinak též nazývány gradientní metody. Jsou založeny na myšlence, že v místech hran dochází ke skokové změně intenzity a jejich derivace v těchto místech nebude mít hodnotu nula, jak je tomu v místech bez hran. Využívají výpočtů gradientu (kolmý vektor k vektoru udávající směr hrany) - jeho velikosti a směru. U metod tohoto typu dochází k problému příliš tlusté hrany – hrana je tak tlustá, že ji tvoří hned několik přímek a nevíme tak kudy přesně vede.

Metody založené na druhé derivaci

Zde jsou hrany definovány jako místa kde je největší první derivace a zároveň druhá derivace prochází nulou. Tyto metody mají oproti předchozím větší časovou náročnost a jsou také více citlivé na šum.

Cannyho hranový detektor

Hon za nejvhodnějším hranovým detektorem probíhající v osmdesátých letech ukončil roku 1986 J.F.Canny se svou prací [15] ve které stanovuje požadavky ideálního hranového detektoru jako:

- minimální chybovost detekce hran,
- detekovaná hrana musí být co nejbližší skutečné,
- každá hrana musí být detekovaná pouze jednou.

Obecně je tento detektor považován za jeden z nejlepších detektorů hran vůbec a je také velmi hojně využíván. Aby byly splněny všechny výše uvedené podmínky, není možné realizovat algoritmus v pouhém jediném výpočtu. Stává se z více kroků, mezi něž patří[22]:

1. Odstranění šumu - většinou prováděno Gaussovým filtrem, což je konvoluční filtr, který využívá váženého součtu hodnot okolních bodů.
2. Určení gradientu - vypočten pomocí konvoluce se Sobelovým operátorem nebo pomocí derivace Gaussovy funkce.
3. Nalezení lokálních maxim - z předchozího kroku jsou vybrána lokální maxima, tím zajistíme, že hrana bude detekována v místě největšího gradientu.

4. Eliminace nevýznamných hran - pomocí prahování s hysterezí, kdy jsou použity dva prahy. Postupně jsou procházeny všechny hrany a je-li gradient bodu vyšší než horní práh je označen jako významná hrana, je-li menší než spodní práh je vyřazen. Pokud spadá do rozmezí takto ohraničených hodnot a některý z bodů v jeho okolí je označen hranou pak i tento bod je hranou.

2.3.6 Možnosti přisvícení

Při použití kamer u parkovacích asistentů je zcela zřejmý požadavek na funkčnost v jakékoliv části dne a při jakýchkoliv podmínkách. Osobní automobily sice jsou vybaveny parkovacími světly, ale jejich osvětlení prostoru nebývá vždy dostačující, proto se možnost přisvícení stává přímo nutností. V úvahu připadají dvě varianty softwarové a hardwarové řešení. **Softwarově** je přisvícení možno provést jak vhodnou úpravou grafického výstupu, tak i přímo nastavením parametrů kamer - vhodným vyvážením hodnot jasu, kontrastu, atd. Druhou možností je **hardwarové** řešení, kdy je nutno přidat další modul například s LED či infračervenými diodami. Popřípadě, i jako nejvhodnější možnost, použít kamery, které tyto prvky obsahují.

Kapitola 3

Navrhované řešení

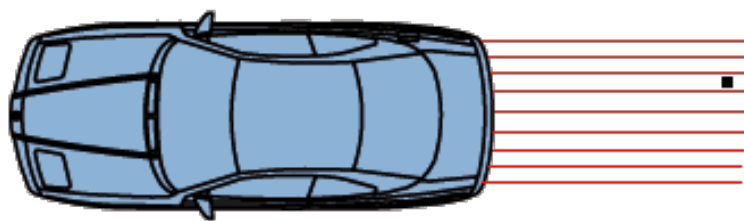
V této kapitole se dozvíme, jaké bylo navrženo řešení a proč tomu tak je.

3.1 Senzory pro měření vzdálenosti

Po důkladném zhodnocení parametrů jednotlivých senzorů bylo možno volit ze dvou typů, a to ultrazvukových či optických. Ostatní senzory byli vyloučeny z důvodu jejich velmi úzkého rozsahu či specifických nároků na vlastnosti materiálu a prostředí ve kterém by mělo měření probíhat. V reálném provozu by nebylo absolutně možné zaručit dodržení těmito senzory vyžadovaných podmínek a nebylo by tak možné zaručit ani jejich funkčnost.

3.1.1 Optické senzory

Při volbě optických senzorů připadaly v úvahu dva typy - laserové či infračervené, založeny na měření intenzity odraženého paprsku. Protože oba senzory se pro náš účel víceméně neliší je zbytečné je hodnotit zvlášť. Oba dva typy vyhovují, co se týče rozsahu měření a vyjma některých materiálů i univerzálnosti použití. Hlavním problémem těchto senzorů je jejich minimální záběr, kdy v podstatě snímají pouze body a nikoliv celý prostor za automobilem viz 3.1.



Obrázek 3.1: Příklad použití optických senzorů

Pokud bychom chtěli pokrýt celou šířku automobilu bylo by potřeba velké množství těchto senzorů a stejně bychom tak pokryli prostor pouze v úrovni senzorů. Je tedy vysoce pravděpodobné, že by velké množství překážek nebylo vůbec zaznamenáno. Ano, teoreticky by bylo možné použít optické senzory s pohyblivými zrcadly jako 3.2, kdy dochází k rozptýlení paprsku a je tak zajištěn široký záběr, ale toto řešení se nejvíce jeví jako v praxi použitelné.

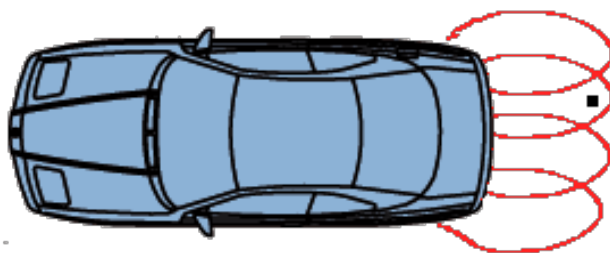
A to převážně kvůli rozměrům těchto senzorů a jejich nepraktickému/nemožnému dodatečnému zabudování do automobilu. Pro představu, rozměry senzoru na obrázku 3.2 jsou 156 mm x 155 mm x 210 mm s váhou pohybující se řádově v kilogramech. Proto nebývají u komerčně vyráběných parkovacích asistentů využívány.



Obrázek 3.2: Laserové měřicí zařízení firmy SICK, převzato z [4].

3.1.2 Ultrazvukové senzory

Jako daleko vhodnější se jeví využívat ultrazvukové senzory, které mají dostačující prostorový záběr a k zabránění celého prostoru za automobilem jich, v závislosti na jejich vlastnostech, stačí řádově jednotky, jak je ilustrováno na obrázku 3.3. V komerční sféře se jedná o nejpoužívanější a v podstatě téměř jediný využívaný typ senzorů. V praxi se zpravidla využívá čtyř čidel na zabránění šířky automobilu. Jejich výhody a nevýhody jsou konkrétněji popsány v kapitole 2.2.4.



Obrázek 3.3: Příklad použití ultrazvukových senzorů

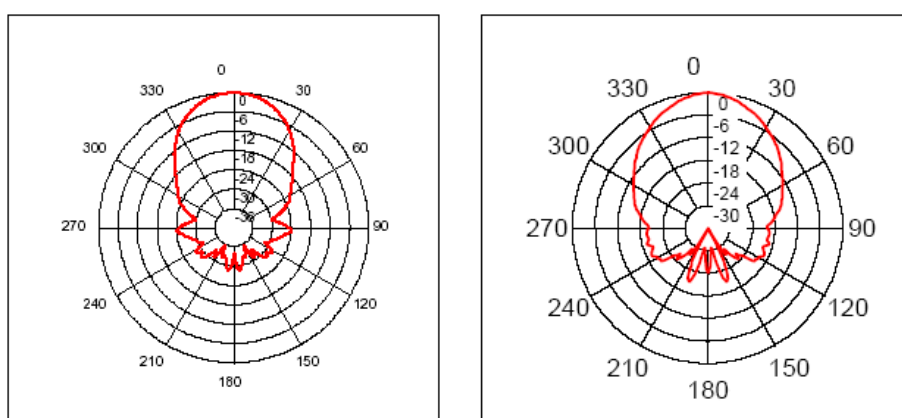
3.1.3 Ultrazvukový senzor SRF08

Pro finální realizaci byl vybrán ultrazvukový senzor SRF08 (obrázek 3.4) od firmy Devantech. V úvahu připadala, krom tohoto senzoru, ještě jeho vylepšená varianta a to konkrétně SRF10. Ta má oproti SRF08 o celých 17 stupňů širší záběr viz vyzařovací charakteristiky 3.5. Hlavní výhodou SRF08 bylo, že na tomto senzoru je navíc zabudován i senzor osvětlení, kterým SRF10 nedisponuje. Ideálním řešením by bylo využít čidel SRF10 a senzor osvětlení řešit samostatně. Z důvodu nedostupnosti dostatečného množství čidel SRF10 pro pokrytí celé šířky automobilu ani použitelného senzoru osvětlení jsem se rozhodl použít pouze čidel SRF08.

SRF08 komunikuje prostřednictvím sběrnice I2C viz 3.4.2. Defaultně je vyroben s adresou 0xE0 tuto adresu je možné však změnit (viz 4.3.5) na jednu z 16 adres. Abychom byli schopni se zařízením komunikovat, je nutno mu přidělit na sběrnici unikátní adresu.



Obrázek 3.4: Ultrazvukový senzor SRF08, převzato z [2].



Obrázek 3.5: Vyzařovací charakteristika SRF08 (levý obrázek) a SRF10, převzato z [9].

Z toho požadavku plyne, že můžeme využít až 16 dálkoměrů současně, což je i pro budoucí rozšiřování parkovacího asistenta o další senzory více než dostatečné.

Registry

SRF08 obsahuje celkově 36 registrů, pro naše potřeby nám si vystačíme se spodními čtyřmi tj registry 0 až 3. Ve zbylých registrech jsou hodnoty odrazů od dalších překážek, což pro naše účely není důležité. Zapisovat je možno do registrů 0, 1 a 2.

Registr 0 označený jako příkazový registr, slouží k zahájení měření. Jeho čtením zjistíme verzi softwaru.

Registr 1 zápisem do tohoto registru nastavujeme maximální zesílení, čtením pak obdržíme hodnotu osvětlení.

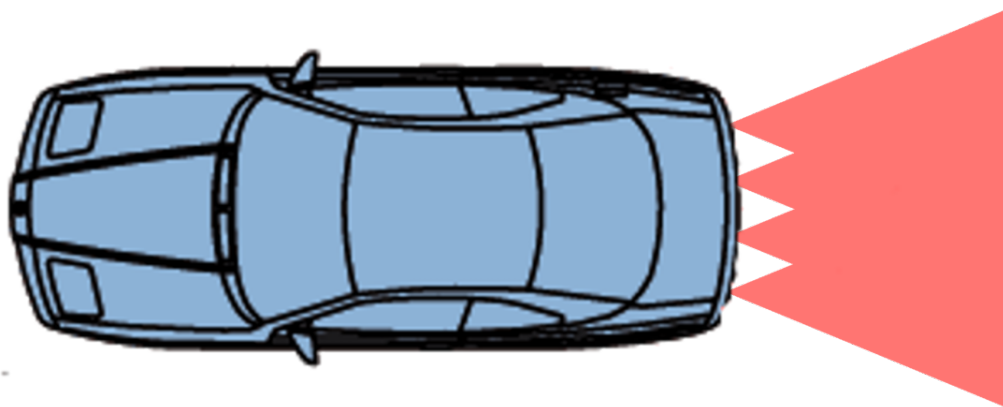
Registr 2 jeho čtením získáváme vyšší byte vzdálenosti k nejbližší překážce, zápisem do tohoto registru lze měnit rozsah měření.

Registr 3 do tohoto registru není možno zapisovat, lze z něj pouze číst a to nižší byte vzdálenost k první překážce.

Zóny

Vzhledem k použití ultrazvukových senzorů je nutno si něco říci o jimi snímaném prostoru. Při použití jednoho senzoru je snímaný prostor zřejmý, vychází totiž z vyzařovací charakteristiky (obr. 3.5) použitého senzoru. Při použití více než jednoho senzoru nastávají některé specifické problémy, jako je například tzv. mrtvá zóna viz. dále. Obecně můžeme snímaný prostor rozdělit do tří základních zón na onu mrtvou zónu. Na zónu, kde probíhá měření a na oblast mimo dosah snímače.

Mrtvá zóna Jedná se o oblast, která vzhledem k vyzařovací charakteristice a rozestupům mezi čidly, není zcela pokryta záběrem čidel. Není zde možné podávat přesné a v některých případech vůbec žádné výsledky. Velikost této oblasti je závislá na vyzařovacím úhlu daného senzoru a na vzdálenosti mezi jednotlivými čidly. Příklad pro SRF08 při použití čtyř čidel a rozestupech 30cm je uveden na obrázku 3.6. Rozestup vychází z předpokladů použití čtyř čidel a průměrné šířky automobilu, která se pohybuje kolem 1,5 metru (experimentálně ověřeno na Škodě Octavia 1.9TDi).



Obrázek 3.6: Použití čtyř čidel SRF08 na osobní automobil.

Pro náš způsob komunikace bylo navrženo užití tří základních zón, avšak v trochu jiném významu než jak je uvedeno výše.

Zóna I. je chápána jako bezpečný prostor. Zóna byla stanovena na vzdálenost od 1,5 metru dále. Horní hranici není třeba omezovat, ta pro naše účely nemá podstatný význam a je dána druhem použitého čidla a jeho nastavením.

Zóna II. pro tuto zónu byla vymezena oblast od 0,5 metru do 1,5 metru. Jedná se o zónu, kde ještě nehrozí akutní nebezpečí, avšak je nutno dbát zvýšené opatrnosti.

Zóna III. je považována za nejrizikovější, protože se vyskytujeme ve vzdálenosti menší než 0,5 metru. Důležité je upozornit, že v rámci této zóny se nachází výše zmiňovaná mrtvá zóna, kde již není možné zaručit plnou funkčnost. Obecně se v této zóně doporučuje couvací manévry ukončit či provádět s maximální opatrností.

3.2 Kamery

Při návrhu kamerového systému bylo rozhodnuto využít tři kamery současně, kdy jedna slouží jako kamera centrální a dvě jako kamery postranní. Centrální kamera by byla umístěna ve středové části nárazníku popřípadě v místě státní poznávací značky, aby snímala co největší část prostoru za automobilem. Postranní kamery by byly umístěny v místech zpětných zrcátek a doplňovaly by tak výhled po obou stranách. Jako prostředků pro realizaci kamerového systému bylo rozhodnuto využít webových kamer. Tyto kamery mají několik základních výhod, zřejmě tou největší je jejich cenová dostupnost. V dnešní době již není problém sehnat kvalitní kamery s Full HD rozlišením za rozumnou cenu. Další výhodou je jejich komunikační rozhraní. Webové kamery zásadně využívají pro svou práci rozhraní USB, které slouží nejen k přenosu dat ale i k napájení. Nevýhodami tohoto typu kamer jsou pak rozměry a různá kvalita u různých výrobců. Pro naši realizaci konkrétně využijeme kamer uvedených v následujících podkapitolách.

3.2.1 Centrální kamera

Jako centrální kamera byla využita **Microsoft LifeCam Studio Win** [3], konkrétněji jde o model Q2F-00004 (obrázek 3.7). Zvolena byla hlavně pro její širokoúhlý záběr, který na dnešním trhu webových kamer těžce hledá konkurenci. Použitý senzor je založený na



Obrázek 3.7: Microsoft LifeCam Studio Win, převzato z [3].

CMOS technologii (2.3.2) a dokáže zabrat v diagonálním směru úhel 75 stupňů. Rozlišení je možno volit z několika variant, největší rozlišení, které je možno využít je 1920 x 1080 se snímkou frekvencí až 30 snímků za sekundu (30 FPS). Tato kamera je také vybavena funkcí digitálního zoomu, který však v naší realizaci, na rozdíl od funkce automatického ostření, nevyužijeme. Automatické ostření probíhá v rozmezí od 10 centimetrů do 10 metrů.

3.2.2 Postranní kamery

Z hlediska snímání prostoru je stěžejní centrální kamera, proto u postranních kamer nejsou kladeny tak vysoké nároky. Z důvodu snížení zatížení výpočetní techniky a přenosových kanálů, bylo dokonce žádoucí použít některé z méně náročných kamer. Na tyto kamery byly využity běžně dostupné kamery s rozlišením 640x480. Snímková frekvence byla u těchto kamer udávána 20FPS. Konkrétně byly použity modely Logitech QuickCam Messenger [10], model V-UBC40 (obr. 3.8) a jako druhá Creative Live! Cam Vista IM [11], model VF0260 (obr. 3.9).



Obrázek 3.8: Logitech QuickCam Messenger, převzato z [10].



Obrázek 3.9: Creative Live! Cam Vista IM, převzato z [11].

3.2.3 Možnosti přisvícení

Při řešení problémů s přisvícením bylo z důvodu omezených možností dodání dalšího zdroje napájení zvoleno jako nejvhodnější řešení softwarové přisvícení založené na grafické úpravě výstupního obrazu z kamer. Funkce přisvícení bude možno spouštět i automaticky na základě hodnot ze senzorů přisvícení umístěných na sonaru SRF08. Osobně se však domnívám, že přisvícení nebude v některých případech vůbec třeba, protože použitá kamera viz 3.2.1 podává, díky jejímu softwarovému řešení, velmi kvalitní obraz i při malém množství osvětlení.

3.3 Uživatelské rozhraní

Při návrhu uživatelského rozhraní jsem se snažil držet několika zásad:

1. Seskupení, jednotlivé prvky rozhraní by měli byti logicky seskupeny.
2. Přehlednost, uživatel by neměl zbytečně pátrat po funkcích, či po tom co znamená která hodnota.
3. Méně znamená více, tento bod částečně souvisí i s bodem předchozím, v tom významu jde však více o grafické ztvárnění. To by nemělo odvádět pozornost uživatele. Tento bod také souvisí s bezpečnostním rizikem. Stále jde o osobní automobil, kde je potřeba věnovat jeho ovládání značnou pozornost, což by při použití nadměrně poutavé grafiky (animace, apod.) mohl být problém, který by mohl vést v krajním případě až ke kolizi.

3.3.1 Prvotní návrh uživatelského rozhraní

Při prvotním návrhu (obr. 3.10) jsem se snažil držet zásad stanovených v úvodu této kapitoly. Rozhodl jsem se proto využít jeden centrální prvek pro seskupení jak hodnoty vzdálenosti tak i různého nastavení. V rámci přehlednosti jsem do příslušných rohů umístil obrazy z jim odpovídajících kamer. Ty jsou navrženy tak, aby se po klepnutí zobrazila v plné velikosti a byl tak zobrazen pouze obraz z vybrané kamery. To je vhodné hlavně v prostorech, kdy se snažíme zaparkovat co nejbližší k předmětu, který se vyskytuje pouze na jedné straně.



Obrázek 3.10: Prvotní návrh uživatelského rozhraní.

3.3.2 Finální uživatelské rozhraní

Po fázi návrhu přišla fáze zhodnocení. Můj návrh jsem prezentoval několika respondentům. Vzhledem k jejich počtu to není možné považovat za seriózní výzkum. Ale jelikož mezi nimi byla zastoupena v podstatě celá šířka populace, můžeme si z něj udělat značnou představu o požadavcích koncových uživatelů. Mezi respondenty byl zastoupen řidič z povolání, automechanik, několik běžných řidičů, řidička i jeden tzv. sváteční řidič. Z tohoto malého průzkumu vyplynulo, že řidiči mají trochu jiný pohled na věc. Jako největší problém se ukázalo tlačítko nastavení, které v podstatě odporovalo i požadavkům, které jsem si sám stanovil. U většiny respondentů již samotné slovo „nastavení“ vyvolávalo jistou nechuť, proto bylo z návrhu zcela odstraněno a jednotlivé funkce, které byly plánovány umístit do tohoto nastavení, byly umístěny pod samostatné ikony.



Obrázek 3.11: Konečná realizace uživatelského rozhraní.

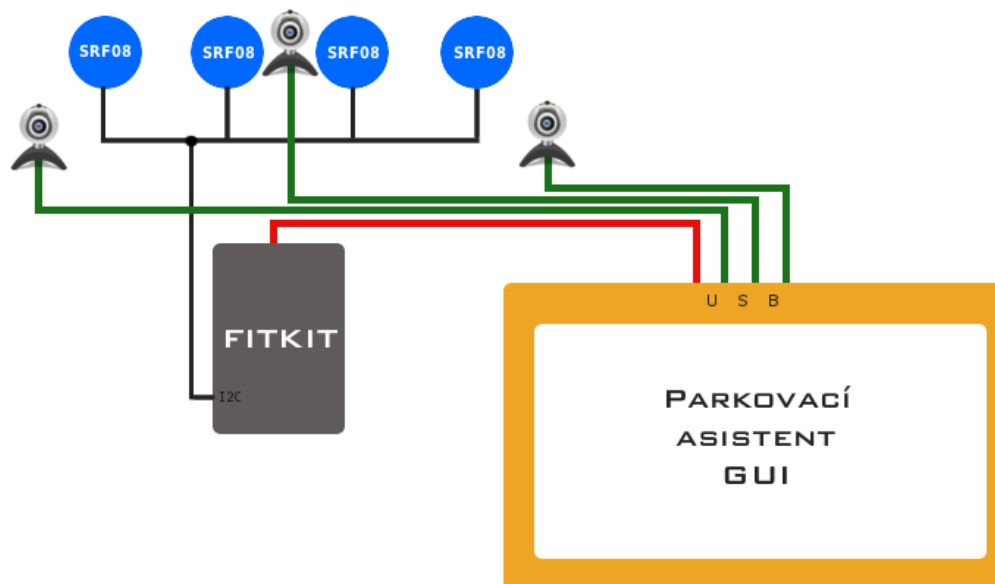
Do návrhu se tak dostala ikonka pro aktivaci/deaktivaci detekce hran. U implementace přisvícení bylo původně uvažováno o pouhé automatické funkci, která by se aktivovala či deaktivovala automaticky dle naměřených hodnot, to se ukázalo jako ne zcela vyhovující a proto byla doplněna možnostmi manuálního zapnutí a vypnutí. Funkce přisvícení byla rovněž umístěna pod samostatnou ikonku.

Zásadní změnu také zaznamenala část pro signalizaci vzdálenosti. Původní myšlenkou bylo zobrazovat pouze vzdálenost k nejbližšímu předmětu, protože ostatní hodnoty signalizují překážky vyskytují se až za první a z tohoto hlediska jsou irelevantní. Vyskytly se však požadavky na informace na které straně se překážka vlastně nachází, proto byla hodnota k nejbližší překážce doplněna hodnotami zbývajících čidel. Vznikla tak dvouřádková varianta, kdy na prvním řádku je zobrazována hodnota z čidla, které udává nejbližší vzdálenost (dále označováno jako centrální hodnota). A v dalším řádku čtveřice všech naměřených hodnot v pořadí senzorů levý, levý prostřední, pravý prostřední a pravý.

Po stranách centrální hodnoty jsou umístěny dva páry grafických signalizačních prvků. Ty jsou postupně zobrazovány v závislosti na vzdálenosti k nejbližší překážce. Rozhodl jsem se použít tři základních barev - zelené, červené a žluté. Jejich použití není náhodné, cíleně jsem zvolil tyto barvy pro podobnost s barevnou kombinací semaforů, kde je již barevná kombinace léty zažitá a jejich významy jsou známy celé populaci. Zelená je použita pro zónu I, tedy signalizující „volno“. Žlutá má reprezentovat oranžovou ze semaforu, tedy „pozor“, proto je využita pro zónu II. A červená, obecně vnímána jako výstraha pro zónu I.

3.4 Schéma komunikace

Tato část se zabývá návrhem komunikace mezi jednotlivými elementy systému (čidel, kamer, pc) mezi sebou. Schématický návrh komunikace je vyobrazen na 3.12.



Obrázek 3.12: Schéma komunikace

Čtveřice ultrazvukových senzorů je řízena pomocí FITkitu více v 3.4.1, kde je připojena na sběrnici I2C (3.4.2), skrze něj jsou rovněž čidla připojena k napájení. O této problematice si blíže povíme v implementační kapitole 4.2.2. FITkit je poté spolu s trojicí kamer připojen ke sběrnici USB (3.4.3). Celý parkovací asistent je řízen pomocí notebooku, který zároveň slouží i jako zobrazovací zařízení.

3.4.1 FITkit

Fitkit [31] je vývojová platforma využívaná na Fakultě informačních technologií, Vysokého učení technického v Brně k výuce hardwarových předmětů. Tento kit je založený na mikroprocesoru řady MCU rodiny MSP430 (Texas Instruments). Z našeho pohledu je nejdůležitější přítomnost sběrnice I2C 3.4.2, přes kterou komunikují ultrazvukové senzory a možnosti využít napájení senzorů prostřednictvím k tomu vyvedených pinů.

3.4.2 Sběrnice I2C

Sběrnice I2C (I2C-bus, Inter-IC-bus, TWI) je dvou vodičové datové propojení mezi jedním a více zařízeními, kterým umožňuje jejich komunikaci. Vodiče jsou nazývány SDA (sériová datová linka) a SCL (hodinový signál). Adresa i data se přenášejí skrz sériovou datovou linku SDA. Připojená zařízení se dělí na master a slave. Master slouží k řízení komunikace a také generuje hodinový signál. Slave je zařízení adresované masterem. Komunikace skrz sběrnici I2C je poloduplexní – na sběrnici může být v jeden okamžik pouze jedno vysílací zařízení.

Všechna zařízení jsou připojeny na jednu sběrnici, přistupovat k určitému z nich lze pomocí jeho unikátní adresy. [28]

3.4.3 Sběrnice USB

Sběrnice USB (Universal Serial Bus) jak název napovídá je univerzální sériová sběrnice, která umožňuje komunikaci nepřebornému množství zařízení. Jedná se o externí sběrnici, která je zajímavá tím, že se vždy jedná o spojení point-to-point, tedy jedním kabelem jsou vždy propojena jen dvě zařízení. Tím se dost liší od klasického použití sběrnice, kde se využívá společných vodičů. V dnešní době jsou touto sběrnici osazeny prakticky všechny osobní počítače.

Kapitola 4

Implementace

V této kapitole si povíme jakým konkrétním způsobem byla řešena programová část této práce a jakých prostředků k tomu bylo využito. Stěžejní část práce zabývající se tvorbou rozhraní a celkovým řízením byla napsána v jazyce C++ s využitím knihoven QT, OpenCV a Libkitclient. Pro samostatnou práci s čidly byl využíván jazyk C a knihovny využívané u platformy FITkit.

4.1 Použité prostředky

V této části si uvedeme, co bylo při implementaci použito za prostředky a knihovny. Uvedeme jejich stručný popis a jelikož všechny použité knihovny jsou šířeny zdarma uvedeme i odkazy, kde je možné knihovny získat.

4.1.1 Qt

Qt je svobodný, multiplatformní framework, mimo jiné, pro vytváření programů s grafickým uživatelským rozhraním. Qt samotné je napsané v jazyce C++, avšak je dostupné i pro jiné jazyky jako například Python, Ruby, Perl atd. V době psaní tohoto textu byla aktuální verze 4.7, tato verze byla také použita při implementaci.

Qt není pouze samotná knihovna, součástí projektu je i celé vývojové prostředí nazývané QtCreator. Kompletní sada nástrojů pro vývoj se pak nazývá Qt SDK a je volně dostupné na stránkách projektu [\[33\]](#). Toto SDK obsahuje, kromě Qt knihovny, také vývojové prostředí QtCreator, vývojové nástroje pro Qt, simulátor pro mobilní telefony a další.

QtCreator je, dle mého názoru, velmi kvalitní vývojové prostředí pro programování aplikací využívající kombinace jazyka C++ a Qt. Obsahuje editor s kompletní podporou jazyka C++, možností doplňování, využívání nápovědy přímo z editoru a další. V rámci QtCreatoru je integrován i QtDesigner (viz. dále) či **QtAssistant** – propracovaný systém nápovědy v kombinaci se vzorovými příklady. Součástí je i zakomponovaný debugger.

QtDesigner je nástroj pro návrh uživatelského rozhraní. Umožňuje tvorbu grafických rozhraní velmi jednoduchým způsobem – přetahováním jednotlivých prvků (widgetu) z panelu do nově navrhovaného okna a zde s nimi dále pracovat. K takto navrženému rozhraní, jde poté pomocí editoru velmi jednoduše přistupovat a značně tak usnadňuje práci.

4.1.2 OpenCV

OpenCV(Open-source Computer Vision) [32] je open-source knihovna pro manipulaci s obrazem. Optimalizována především pro práci s počítačovým viděním v reálném čase. Obsahuje více než 2500 optimalizovaných algoritmů počítačového vidění. Je napsána v jazycích C a C++ a je dostupná jak pro Windows, tak i pro Linux a Mac OS a to pod licencí BSD. V době psaní této práce byla nejnovější verze 2.3.1, ta je také použita. Na stránkách projektu [32] je nejen dostupná ke stažení, ale také je zde k ní zpracována kvalitní dokumentace. Při programování mi také velmi pomohla kniha [23].

Struktura

OpenCV je strukturováno do několika modulů:

- Core – zde jsou definovány datové struktury a základní funkce využívané ostatními moduly.
- Imgproc – modul pro zpracování obrázků – filtry, transformace, converze, histogramy, atd.
- Video – analýza videa jako například algoritmy pro odhad pohybu, sledování objektů, apod.
- Objdetect – modul pro detekci objektů z předdefinovaných tříd například pro oči, obličeje, lidi, atd.
- Highgui – rozhraní pro práci se zachytáváním videa, práci s obrázky a prvky pro tvorbu jednoduchého uživatelského rozhraní.
- A další modely pro práci s 3D, GPU atd, více se dozvíte na stránkách projektu [32] nebo v knize [14].

4.1.3 Libfitkit

„Knihovna libfitkit zastřešuje základní funkce pro použití MCU a komunikaci MCU s komponentami na FITkitu nebo PC“, citováno z [27]. Při jejím použití je nutné definovat pro MCU, kromě vlastních funkcí, ještě tři další, které jsou volány a to `fpga_initialized` pro inicializaci FPGA, `print_user_help` pro výpis uživatelské nápovědy a `decode_user_cmd` pro dekodování uživatelských příkazů. Kromě toho obsahuje řadu funkcí pro vypisování textů do terminálu, různé inicializační funkce a některé uživatelské proměnné. Více detailů a mnoho dalších funkcí je uvedeno na stránkách knihovny [27]. Knihovnu je možné získat z SVN repozitáře projektu na téže stránkách.

4.1.4 Libkitclient

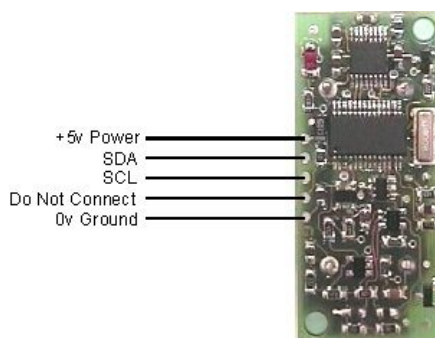
Libkitclient je další z knihoven pro komunikaci s FITkitem. Je napsána v jazyce C++ a poskytuje jednotné API pro interakci s FITkitem, jak v prostředí Windows tak i pod systémem Linux. Umožňuje přímou komunikaci skrz naši aplikaci. Dokumentace a další informace [29].

4.2 Zapojení

Před samotným programováním je nutné mít vše správně připojeno.

4.2.1 Zapojení SRF08

Ultrazvukový senzor je vybaven pětící vývodů, viz. obrázek 4.1. Je zde vývod, postupně, pro napájení (vyžadováno +5V), následované vodiči sběrnice I2C – sériovou datovou linkou (SDA) a hodinovým signálem (SCL). Čtvrtý vývod (označený „Do Not Connect“) musí zůstat nezapojen – jedná se o vývod MCLR procesoru a využívá se pouze ve výrobě k na-programování. Posledním vývodem je zem (0V). Vodiče SDA a SCL musí mít někde na sběrnici I2C umístěny zdvihadí odpory (tzv. pull-up rezistory), těmi je však FITkit přímo vybaven, takže není potřeba žádné dodatečné řešení.



Obrázek 4.1: Vývody SRF08, převzato z [13].

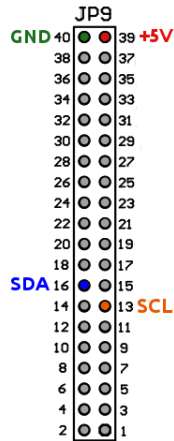
4.2.2 Připojení SRF08 k FITkitu

Jak jsme si již řekli, potřebujeme napájecí napětí +5V, zem (0V) a vývody sběrnice I2C. Všechny tyto požadavky splňuje rozhraní JP9 (znázorněno na obr. 4.2) v levé dolní rohu FITkitu. Zde je +5V vyvedeno na pinu 39, 0V(zem) na pinu 40. Sběrnice I2C je vyvedena na piny 16 (SDA) a 13 (SCL). Nutno říci, že tento popis se týká použití na FITkitu verze 2.0 s mikrokontrolerem MSP430F2617.

Připojení by nemělo dělat větší problémy. Zvýšenou pozornost je nutné věnovat pouze vodičům mezi FITkitem a senzory. Při použití méně kvalitních, nestíněných vodičů, či nedokonalé vodivě spojených vodičů, docházelo k vzájemnému rušení a tak k problémům s komunikací.

4.2.3 Kamery

Připojení kamer ke sběrnici USB je zcela běžná věc, kterou není potřeba nijak rozepisovat. To v dnešní době zvládá opravdu každý. Zde bych jen upozornil na problémy se staršími typy webových kamer. Některé starší modely totiž nedisponují možností sdílení sběrnice a zabírají si pro sebe celou její šířku, což může při použití USB hubu činit problémy. Řešením je využít pro každou kameru jinou sběrnici (jiný port počítače). U novějších modelů takovéto problémy nejsou známy.



Obrázek 4.2: Označení vývodů na portu JP9.

4.3 Práce se senzory SRF08

Programová část pro práci s ultrazvukovými senzory je rozdělena do několika modulů. V souboru `main.c` je hlavní řízení komunikace, v souboru `cidla.c` jsou funkce pro práci s čidly a v souboru `i2c.c` funkce pro práci přímo se sběrnici I2C. Před prvním použitím senzorů je třeba provést nastavení jeho adresy (4.3.5), to se provede odkomentováním funkce `srf08_change_adress`.

Komunikace probíhá následovně:

1. Inicializace FITkitu, zastavení watchdogu, nastavení diod.
2. Inicializace I2C sběrnice.
3. Inicializace SRF08 senzorů.
4. Nekonečná měřicí smyčka:
 - (a) Start měření.
 - (b) Vyčtení hodnot vzdálenosti a osvětlení.
 - (c) Zapsání hodnot do terminálu.
 - (d) Vyčkání 70ms před započítáním dalšího měření.

4.3.1 Práce se sběrnici I2C

Funkce pro práci se sběrnici jsou obsaženy v souboru `i2c.c`, tento soubor byl převzat z práce [26]. Obsahuje funkce pro inicializaci `i2c_init`, zápis `i2c_write` a čtení `i2c_read` z této sběrnice. Tyto funkce jsou zde, jak pro FITkit verze 1.x, tak i pro 2.0.

4.3.2 Inicializace SRF08

Inicializace senzorů probíhá ve funkci `srf08_init` ze souboru `cidla.c`. Při inicializaci se nastavuje zápisem do registru 1 hodnota maximálního zesílení a do registru 2 rozsah měření.

Kompletní popis registrů jsme již uvedli v [3.1.3](#). Maximální analogové zesílení je nastaveno na minimální hodnotu 94, do registru 1 tak zapisujeme hodnotu 0. Analogové zesílení snižujeme z důvodu, abychom se vyvarovali zachycení odrazů z předchozích měření. Rozsah je po startu nastaven na hodnotu 11 metrů, což vůbec neodpovídá skutečnému dosahu SRF08, který je 6 metrů [\[13\]](#). Nastavení této hodnoty určuje i dobu po kterou senzor čeká na odražený signál, proto je při inicializaci do registru 2 zapsána hexa hodnota 0x5D, která snižuje měřený rozsah na 4 metry, což je pro funkci parkovacího radaru dostatečné.

4.3.3 Start měření

Měření je zahájeno funkcí `srf08_start`, která zapíše do registru 0 hodnotu 0x51, která značí, že požadujeme výsledek v centimetrech a tím je zahájeno měření.

4.3.4 Měření vzdálenosti

Výsledek měření je dostupný po 65ms v registrech 2 (vyšší byte) a 3 (nižší byte) odkud je čteme pomocí funkce `srf08_cm`, která jako výsledek svého volání, vrátí buď vzdálenost nebo chybovou hodnotu. Zároveň s měřením vzdálenosti je změřena i hodnota ze senzoru osvětlení, tu čteme z registru 1 funkcí `srf08_light`. Senzor není přesně kalibrován a tak je hodnota osvětlení spíše orientační – ve tmě je hodnota 2-3 a při jasném světle přes 240 [\[13\]](#).

4.3.5 Adresa senzoru

Jak jsme si již řekli v kapitole o senzoru SRF08 ([3.1.3](#)), aby bylo možné komunikovat na jedné sběrnici naráz s více senzory, je nutné, aby každý měl unikátní adresu. SRF08 je z výroby naprogramována na adresu 0xE0. Zda byla přeprogramována a jaká je adresa zjistíme z počtu blikání LED diody na senzoru více v dokumentaci [\[13\]](#). Adresu lze změnit pošleme-li do registru 0 následující sekvenci 0xA0, 0xAA, 0xA5, následovanou adresou. Pro tuto možnost najdete v adresáři pro FITkit soubor `cidla.c`, kde je vytvořena funkce `srf08_change_adress`, která provede změnu adresy. Adresu lze změnit na jednu z následujících 16-ti adres E0, E2, E4, E6, E8, EA, EC, EE, F0, F2, F4, F6, F8, FA, FC nebo FE. Před změnou adresy je nutné, aby na sběrnici bylo přítomno pouze jedno jediné zařízení.

4.3.6 Tisk hodnot do terminálu

Po změření hodnot jsou jednotlivé hodnoty posílány funkcí `term_send_str` z knihovny `lib-fitkit` ([4.1.3](#)), na terminál v následujícím tvaru:

`LIGHT|L|ML|MR|R=MIN>`, kde:

- `LIGHT` je nejmenší hodnota osvětlení, která byla z čidel naměřena.
- `L`, `ML`, `MR` a `R` jsou hodnoty vzdálenosti naměřené z příslušejícího čidla - levého, prostředního levého, prostředního pravého a z čidla umístěného na pravé straně.
- `MIN` potom udává nejmenší hodnotu vzdálenosti, která byla naměřena.
- Celá sekvence je ukončena úhlovou závorkou, aby bylo možné data rozdělit na jednotlivé sekvence.

4.4 Přenos dat z FITkitu do QT

Čtení dat z FITkitu nám umožňuje knihovna libkitclient (4.1.4). Díky ní je možné ve svém programu otevřít terminál a data z něj postupně vyčítat. Ihned po inicializaci FITkitu proběhne prázdná čtecí smyčka, který běží dokud nenarazí na úhlovou závorku „>“, tím odfiltrujeme všechny inicializační zprávy FITkitu a dostáváme se rovnou k datům. Funkce pro čtení hodnot se nazývá `updateDistance` a díky časovači `timerDistance` je volána v pravidelných intervalech. Jelikož data jsou v tvaru uvedeném v předchozí kapitole, není problém je zpracovat na jednotlivé prvky.

4.5 Kamery v QT

Využití kamerového systému v QT je možné díky knihovně OpenCV (4.1.2), konkrétně jejímu modulu `highgui`. Ten práci s kamerami velmi zjednodušuje a hlavně zefektivňuje. Pro zachytávání obrazu z kamer je využívána třída `VideoCapture`, ta obsahuje i základní funkce pro práci s nimi. Nejdůležitější je funkce `open`, která nám kameru zpřístupní, dále funkce `grab` a `retrieve`, ty slouží k zachycení jednotlivých rámců obrazu a nakonec `release`, která kamery zase uvolňuje. V prvé řadě je důležité si říci, že v různých operačních systémech jsou kamery identifikovány různě. V systémech Windows se využívá číselných hodnot 0 až N podle množství připojených kamer. Funkci `open` poté předáváme identifikátor jednotlivých kamer např. `open(0)` pro otevření kamery nastavené jako defaultní. Naopak v linuxu jsou identifikovány způsobem `/dev/videoN`, kde N opět značí číslo kamery, funkce `open` by v tomto případě vypadala `open(/dev/video0)`. Funkce `grab` a `retrieve` lze nahradit samotnou funkcí `read`, avšak vzhledem k tomu, že čteme obraz ze tří kamer současně je lepší využít předchozí kombinace – `grab` obraz zachytí a `retieve` teprve dekoduje a zpracuje, což je při správném využití mnohem efektivnější a minimalizujeme tak zpoždění obrazu.

4.6 Popis hlavního programu

Hlavní okno aplikace je tvořeno pod třídou `QMainWindow`, která se využívá pro window aplikace. V jejím konstruktoru probíhá inicializace FITkitu díky funkci `initFitkit` a ve funkci `initCam` zase inicializace kamerového systému. Poté probíhá nastavování jednotlivých grafických elementů a hlavně spouštění jednotlivých časovačů, které jsou základem celého programu, protože v pravidelných intervalech vyvolávají funkce pro čtení, jak kamer, tak hodnot vzdálenosti.

4.6.1 Práce s kamerovým systémem

Práce s kamerovým systémem je vykonávána ve funkci `updateCams`, zde probíhá periodické zpracování obrazu ze všech tří kamer současně. Postup je následující:

1. Dekódování obrazu z kamery.
2. V případě zapnuté detekce hran je volána funkce `EdgeDetection` (4.6.2), která vrací obraz s vyznačenými hranami.
3. Pokud je také zapnut noční režim, je vyvolána funkce `ChangeBrightness`, která upraví jas obrazu aby byl lépe viditelný.

4. Jak již bylo zmíněno, OpenCV používá pro svou vnitřní reprezentaci barevný model BGR je nutné jej pomocí `cvtColor` převést na RGB.
5. V dalším bodě RGB obraz převádíme na `QImage`, aby bylo možné jej zobrazit v uživatelském rozhraní
6. Tyto kroky se, vyjma možnosti detekce hran, provádějí současně i pro levou a pravou kameru.

4.6.2 Detekce hran

Probíhá ve funkci `EdgeDetection`, kam je předám obraz zachycený kamerou. V první řadě je volána funkce `CannyAlg`. V které je volána posloupnost funkcí `cvtColor`, `GaussianBlur` a nakonec `Canny`. Všechno jsou to funkce implementované uvnitř OpenCV. `CvtColor` slouží ke konverzi z jednoho barevného prostoru do druhého, v tomto případě z RGB do stupňů šedi (grayscale). Takto převedený obrázek je poté vstupem do funkce `GaussianBlur`, která odstraní šum použitím Gaussova filtru. Nakonec je takto zpracovaný obraz předán funkci `Canny` jako parametr je vstupní a výstupní obrázek, dále hodnota nižšího a vyššího prahu pro prahování a jako poslední je velikost detekčního okna pro Sobelův operátor. Výstupem z těchto funkcí je obraz, který obsahuje pouze hrany na bílém podkladu. Ten je poté pomocí `findContours` prohledán a v obrázku jsou nalezeny obrysy, jelikož mu byl předložen pouze obrázek obsahující hrany na bílém pozadí, jsou nalezeny právě tyto hrany. Ty jsou poté skrz funkci `drawContours` zakresleny přímo do obrazu zachyceného z kamery.

4.7 Popis uživatelského rozhraní

Uživatelské rozhraní je zobrazeno na obrázku 4.3. V něm jsou vyznačeny jednotlivé prvky. Ty si nyní popíšeme.

Levá/pravá kamera Zobrazují obraz z postranních kamer, při klepnutí na příslušnou kameru se její obraz maximalizuje a zobrazí se na místě centrální kamery, ostatní jsou zatím deaktivovány.

Prvek A Aktivuje a poté i deaktivuje zvukovou signalizaci – zvuk je odstupňován podle vzdálenosti různým taktem.

Prvek B Slouží pro zapnutí celoobrazovkového režimu, poté je ikona změněna a složí pro návrat do běžného režimu.

Prvky C Vykonávají funkci grafické signalizace vzdálenosti. Opět jsou závislé na nejmenší vzdálenosti k překážce a podle toho je zobrazen zelený, žlutý či i červený sloupec. Na obrázku jsou pro ilustraci vyobrazeny všechny tři sloupce, při této vzdálenosti však bývá zobrazen pouze zelený sloupec.

Prvek D Označován jako centrální vzdálenost. Je to nejmenší hodnota ze všech naměřených a dle její hodnoty je odvozována zvuková a grafická signalizace.

Prvek E Obsahuje pole naměřených hodnot, které jsou seřazeny dle osazení senzorů na automobilu – postupně levá, levá prostřední, pravá prostřední a pravá hodnota.

Prvek F Slouží k aktivaci a poté opětovné deaktivaci režimu detekce hran.



Obrázek 4.3: Popis uživatelského rozhraní.

Prvek G Slouží k aktivaci přisvětlení, je možné volit ze tří režimů:

1. Auto – aktivuje se automaticky na základě hodnot naměřených čidly.
2. On – přisvětlení je trvale zapnuto.
3. Off – trvale vypnuto.

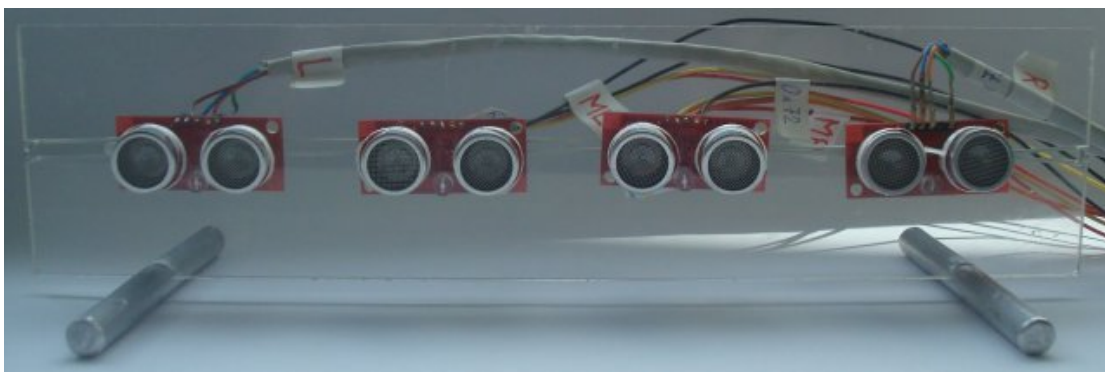
4.8 Přenositelnost

Všechny použité knihovny jsou dostupné jak pro Linux tak i Windows. Po korektní instalaci by neměl být s přenositelností problém. Avšak i přesto by bylo nutné provést zásahy do kódu a to z důvodů různé identifikace jednotlivých kamer, kdy není možné zaručit, že se jednotlivé kamery identifikují na všech systémech ve stejném pořadí, tudíž pod stejným identifikátorem. O rozdílech v identifikaci zařízení mezi systémy Linux a Windows jsme si již řekli kapitole 4.5.

Kapitola 5

Testování a další vývoj

Vytvořený parkovací asistent byl implementován a zprovozněn. Testování probíhalo ve dvou fázích. První bylo vytvoření přípravku pro uchycení senzorů (obr. 5.1). Na tomto přípravku pak probíhalo základní zprovoznění a testování.



Obrázek 5.1: Přípravek pro základní testování.

Jakmile byla vyřešena funkčnost v laboratorních podmínkách, přišla na řadu fáze dvě. V té byly čidla osazeny a testovány na reálném osobním automobilu. Testování proběhlo na automobilu Škoda Octavia. Při testování jsem nenarazil na větší technické problémy. Jako nedostatek se však ukázalo, že zpoždění obrazu z kamer je v praxi mnohem markantnější, než jak se jevílo v laboratorních podmínkách. Přešel jsem tedy k optimalizaci kódu, zlepšení však bylo minimální. Proto bych do budoucna navrhoval otestovat systém s jiným typem kamer, například s některým z kamerových modulů (2.3.4) využívaných v robotice. Popřípadě vyzkoušet využití i jiných programových prostředků, které by poskytovali efektivnější prostředky pro práci s více kamerami současně.

Pro další vývoj bych navrhoval implementovat přenos jak naměřených hodnot, tak i obrazu z kamer bezdrátovým způsobem. Navrženou aplikaci přizpůsobit, popřípadě vytvořit samostatnou verzi, pro běh například na tabletech, chytrých telefonech či nějakým způsobem integrovat do GPS navigací. Tato zařízení by pak byla možno využívat jako zobrazovací prvky asistenta, tím by odpadla nutnost pořizování a instalace dalšího, jinak nevyužitelného, zobrazovacího elementu v automobilu. Použití by také bylo pro uživatele mnohem komfortnější.

Dalším projektem, který by mohl být na základě této práce postaven, by byla implementace tzv. aktivního parkovacího asistenta, který by se aktivně zapojoval do řízení při parkování a v konečné fázi by byl schopný provést řídicí manévry zcela samostatně bez asistence řidiče.

Kapitola 6

Závěr

Cílem této práce bylo navrhnout a zprovoznit parkovacího asistenta. Za tímto účelem byly prozkoumány jednotlivé typy senzorů pro měření vzdálenosti a možnosti použití kamer. Pro realizaci byly užity ultrazvukové senzory v kombinaci s webovými kamerami. Bylo navrženo uživatelské rozhraní, kde byly sloučeny hodnoty z jednotlivých senzorů a kamer. Rozhraní bylo také doplněno o funkci na detekci hran, zvukovou a grafickou signalizaci a možností, nejen automatického, přisvětlení obrazu. Toto rozhraní bylo také implementováno.

Celý systém byl testován na reálném vozidle. Testování ukázalo, že senzory pro měření vzdálenosti nevykazují žádné problémy a jimi naměřené hodnoty, skutečně odpovídají realitě. Slabým místem se ukázal kamerový systém, který je sice plně funkční, avšak obraz není naprosto plynulý a dochází, sice k malému, ale nepříjemnému zpoždění obrazu. Proto bych před případným nasazením do praxe, navrhoval využití jiného typu kamer či knihovny, který umožňuje efektivnější práci s více kamerami současně.

Takto vytvořený parkovací asistent by při dalším vývoji mohl vést, jak je nastíněno v předchozí kapitole, až k tzv. aktivnímu parkovacímu asistentu, který sám (bez zásahu řidiče) provádí parkovací manévry, a to pouze pomocí senzorů.

Literatura

- [1] Zákazníci baží po samočinném parkování od Fordu! [online].
<http://www.ford.cz/AboutFord/News/VehicleNews/2011/zakaznici-bazi-po-samocinnem-parkovani-od-fordu>, 2011-11-16 [cit. 2012-04-14].
- [2] Devantech SRF08 Range Finder [online].
<http://acroname.com/robotics/parts/R145-SRF08.html>, 2011-12-13 [cit. 2012-04-15].
- [3] LifeCam Studio [online].
www.microsoft.com/hardware/cs-cz/p/lifecam-studio/Q2F-00004, 2012 [cit. 2012-04-20].
- [4] Laserová měřicí technika [online].
<https://www.mysick.com/partnerPortal/ProductCatalog/DataSheet.aspx?ProductID=33755#>, 2012 [cit. 2012-05-05].
- [5] Navigační systém s dotykovou obrazovkou a s parkovací kamerou [online].
<http://www.hyundai-i30.cz/>, [cit. 2012-04-10].
- [6] Parkovací asistent PMW-40 [online].
<http://img.seznamzbozi.cz/direct/iR/importprodukt-orig/111/1117dd9611c2ee618c40b5feb690023f.jpg>, [cit. 2012-04-10].
- [7] Parkovací asistent se 4 zadními senzory - černé [online].
<http://www.karavan3nec.cz/elektro/parkovaci-asistent-se-4-zadnimi-senzory/>, [cit. 2012-04-10].
- [8] Jednocestná optická závora [online].
http://www.balluff.cz/bos_principy_definice.asp, [cit. 2012-04-12].
- [9] Ultrasonic Rangers FAQ [online].
http://www.robot-electronics.co.uk/htm/sonar_faq.htm, [cit. 2012-04-15].
- [10] QuickCam Messenger [online]. <http://www.logitech.com/en-us/435/3780>, [cit. 2012-04-20].
- [11] Live! Cam Vista IM (VF0260) [online]. <http://support.creative.com/-Products/ProductDetails.aspx?catID=218&CatName=Web+Cameras&subCatID=846&subCatName=Live!%20Cam%20Series&prodID=15311&prodName=Live!+Cam+Vista+IM+%28VF0260%29> , [cit. 2012-04-26].
- [12] e-CAM50_OMAP35x - 5 MP HD Camera Board for OMAP 35x Processors [online].
http://www.e-consystems.com/5MP_hdcamera_omap.asp, [cit. 2012-04-27].

- [13] SRF08 Ultra sonic range finder - Technical Specification [online].
<http://www.robot-electronics.co.uk/htm/srf08tech.html>, [cit. 2012-04-28].
- [14] Bradski, G. R.; Kaehler, A.: *Learning OpenCV*. O'Reilly, 2008, ISBN 978-0-596-51613-0.
- [15] Canny, J.: A Computational Approach to Edge Detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1986.
- [16] equasys: Color Formats [online]. <http://www.equasys.de/colorformat.html>, 2010 [cit. 2012-04-25].
- [17] equasys: Color Conversion [online].
<http://www.equasys.de/colorconversion.html>, 2011 [cit. 2012-04-25].
- [18] Ing. Filip Orság, Ph.D.: *Robotika, Studijní opora*. FIT VUT, 2006.
- [19] Ing. Josef FORMÁNEK, Ph.D.: Optické senzory pro měření vzdálenosti [online].
http://home.zcu.cz/formanek/mmvyuka/Data/opticke_snimace/optika_vzdalenost.htm, 2007-01-11 [cit. 2012-04-20].
- [20] Kab.: Analogová versus digitální kamera. *Automatizace*, 2004.
- [21] Kočí, M.: *Laserový proximní skener - mechanika: Laser Proximity Scanner - Mechanics*. Diplomová práce, Vysoké učení technické, Fakulta elektrotechniky a komunikačních technologií, 2009.
- [22] Kočí, M.: *Detekce šířky papírární linie u otisku prstu*. Diplomová práce, Vysoké učení technické, Fakulta informačních technologií, 2011.
- [23] Laganière, R.: *OpenCV 2 computer vision application programming cookbook: over 50 recipes to master this library of programming functions for real-time computer vision*. Packt Publishing, 2011, ISBN 978-1-84951-324-1.
- [24] Martínek, R.: *Senzory v průmyslové praxi*. Technická literatura BEN, 2004, ISBN 80-7300-114-4.
- [25] netcam.cz: Obrazové snímače CCD vs. CMOS [online].
<http://www.netcam.cz/encyklopedie-ip-zabezpeceni/obrazove-snimace-ccd-cmos.php>, [cit. 2012-04-27].
- [26] Novotný, T.: *Řízení robota pomocí FITkitu*. Diplomová práce, Vysoké učení technické, Fakulta informačních technologií, 2008.
- [27] Slaný, K.: Knihovna libfitkit [online].
http://merlin.fit.vutbr.cz/FITkit/docs/firmware/mcu_libfitkit.html, 2009-03-20 [cit. 2012-05-03].
- [28] Tišnovský, P.: Komunikace po sériové sběrnici I2C [online].
<http://www.root.cz/clanky/komunikace-po-seriove-sbornici-isup2supc/>, 2009-01-08 [cit. 2012-05-02].
- [29] Vavruša, M.; Vašíček, Z.: Knihovna libkitclient [online].
<http://merlin.fit.vutbr.cz/FITkit/docs/navody/libkitclient.html>, 2009-09-20 [cit. 2012-05-03].

- [30] Vojáček, A.: PMD senzor & 3D měření vzdálenosti - 1. část Princip [online].
<http://automatizace.hw.cz/clanek/2006070301>, 2006-08-01 [cit. 2012-04-15].
- [31] WWW stránky: FITkit. <http://merlin.fit.vutbr.cz/FITkit/>.
- [32] WWW stránky: OpenCV. <http://opencv.willowgarage.com/wiki/>.
- [33] WWW stránky: Qt – Cross-platform application and UI framework.
<http://qt.nokia.com/>.

Příloha A

Obsah DVD

K tomuto dokumentu náleží příloha v podobě DVD, které obsahuje:

- Fotografie z testování a snímky z běhu aplikace.
- Zdrojové soubory pro umístění do adresářové struktury FITkitu.
- Zdrojové soubory aplikace.
- Stručnou instalační příručku.
- Zdrojový tvar písemné zprávy.
- Tuto písemnou zprávu ve formátu PDF.